1981

A Study of the Computer Software Products Industry

Ellis Horowitz
University of Southern California

Robert Hollies
University of Southern California

A Study of the Computer Software Products Industry*

Ellis Horowitz
Computer Science Department
University of Southern California

Robert Hollies
Computer Science Department
University of Southern California

ABSTRACT

In this paper the computer software products industry is defined as consisting of those companies who sell predefined and prepackaged software for execution on the purchaser's medium or large scale digital computer. This industry is currently playing a crucial role, as a severe bottleneck in software development exists. Our goal is to present an analysis of this industry covering a broad range of issues. Our hope is to present a picture which will enlighten both industry consumers and participants concerning the essential forces which govern growth and competition in this arena. To support our synthesis, we have performed an indepth study of over twenty companies and interviewed several industry leaders. Among our contributions is a taxonomy which permits a characterization of the diverse product offerings. Also we present figures which describe the industry in terms of the number of firms, the size of the firms, and the number of firms entering and leaving each year. We show how the firms can be segmented into three groups and discuss interactions among members of different groups. Finally we place the industry on the evolutionary life cycle and conclude with some thoughts about its future.

During the course of this research we found many people who were willing to help us. In particular I would like to thank Larry Walke, President of International Computer Programs Inc. (ICP), Herbert Gepner of Datapro Research Corporation, and Gail Lepard of Computer Sciences Corporation (CSS) for providing valuable resource material. We would also like to thank Walter Scacchi of USC and Lynn Markus of MIT for insightful discussions of this work.

INTRODUCTION

The computer industry is a vital segment of our economy. To understand its changing patterns requires the ability to apply business, economic, and technological methodology in a complex manner. Because software represents a major bottleneck in the further rapid spread of computers, we have chosen to analyze the computer software products industry as a first step. By determining the essential

patterns which govern growth and competition, our hope is that industry participants and consumers will better appreciate the environment in which they find themselves and the roles they are playing.

The data upon which this article is based were gathered in the Fall of 1980. The work was performed in the context of a research seminar given in the Computer Science Department at the University of Southern California. Ten graduate students and one student from the Graduate School of Business comprised the class. Our initial direction was to try to do everything we could to understand the dynamics of the software products industry. We searched virtually all of the available literature, but our primary mechanism for learning about the industry was to perform structured interviews of representatives from software vendors. We began with a list of companies in Southern California and a questionnaire was verified through several test interviews. By the end of the semester we had completed twenty-one company visits. This report is in part a summary of those interviews. Although this sample cannot be interpreted as statistically adequate, we believe that this bias does not detract from our overall conclusions. The observations which we make are based both upon the research data collected and upon the conversations we have had with many people, some of whom have been involved in the industry since its inception.

The computer software products industry, sometimes called the independent packaged software industry, is composed of those companies which sell predefined and prepackaged computer software systems to the computer user community at large. In order to focus the effort as precisely as possible, the decision was made not to include companies which sell turnkey systems, i.e., hardware and software together, nor to include companies which offer software packages solely over their computer network. We also decided not to focus on the emerging field of software for "personal" or small business computers. Thus, the results described here relate primarily to companies which sell software packages for medium and large mainframe computers. On the other hand, for a company to be included in our purview it was not necessary that its entire business be in software products. Many of the companies we studied had other divisions which offered services such as mainframe manufacturing, contract programming, timesharing, facilities management, consulting, etc. In those cases we have concentrated on their software products division.

This paper begins with a discussion of the nature of the software product. We believe that an appreciation of this industry is not possible without a careful understanding of the technical aspects of computer software, especially those factors which influence product and market strategy. The next section, Results of the Interviews, reports the results of our interviews. We group our information into six categories--company background, product lines, methods of advertising, consumer profiles, development practices, and support mechanisms. In the fourth section we present a taxonomy which tries to categorize the industry's rather diverse product offerings. The trade literature continues to refer to systems, utilities, and applications as the three major divisions of products. To us this appears to be totally inadequate. In its place we offer six categories, their definition, and some sample products. In the fifth section, Characteristics of the Industry, we organize vendor firms into three groups and discuss their interactions. We also discuss where the industry is located on the industry life-cycle and the implications this has for further development. The paper concludes with a brief summary.

THE NATURE OF THE SOFTWARE PRODUCT

"Software is not so much a product but a service."

Because the nature of the software product is not widely appreciated, it seems worth the time to discuss it briefly before we launch into a presentation of our findings. At one level of understanding, the product which is being manufactured consists of a computer tape which contains a sequence of computer instructions for performing specific tasks, and documentation which explains how to install and how to use the system created by these instructions. This view is certainly accurate, but it is clearly too simplistic and contributes to misconceptions about ongoing costs of the product.

For example the fact that copies of the tape and the documentation can be made virtually for free leads to the mistaken view that built, only marginal costs are associated with continued sales. Actually the computer software package is in continual need of support, both to eliminate newly discovered errors and to update it. The updates may be caused not by the vendor, but either by the computer hardware manufacturer who alters the operating system, or by the end-user, who adds new hardware to the existing configuration. Studies from the computer science community point to the fact that the cost of maintaining a software system typically exceeds the cost of its development by a factor of two and more (Brooks, 1975). In addition, the cost of sales can be substantial for a product whose benefits are difficult to determine and whose quality is difficult to evaluate. Another possible major expense is the degree of customization which must be applied to the product before it will be purchased. This fact is especially relevant for so-called application software, as purchasers are naturally reluctant to alter their operations to fit the computer. Another expense is the amount of instruction that must be provided before the product can be successfully employed. Though the charge for this instruction is typically passed onto the purchaser, the vendor must be ready to supply the instruction. Collectively these factors imply that a considerable ongoing investment may be required even after the product has been sold.

Another common misconception is that once a better design has been conceived of, all that remains is to produce it and market it adequately to achieve success. This is also not true and for more than one reason. At present it is impossible to create a prototype of a software design to determine if it has potential as a product. Instead the firm must commit its resources to building (virtually) the complete product, without solid proof that it will perform as expected. Even worse, it is very difficult to get an accurate estimate of the time and money it will take to build the product. The computer science field has not matured to the point where dependable methods exist to accurately estimate the costs of a job (Bailey-Basili, 1981). Moreover, if a development effort gets behind schedule placing more people onto the project frequently results in further delays. As a consequence, there are substantial risks associated with large software development. Another inhibiting factor is that a new product which is trying to replace an existing software system, must offer sufficient benefits so that it is worth the buyer's effort to displace the older system. This fact is especially relevant where a high learning curve exists for the new product. These considerations imply that a high front-end investment will be needed before the product can be built, tested, and recognized as superior in some appropriate sense. New companies entering this arena must face and solve this problem first.

In contrast to some industries where a sale, once completed, requires no more atten-

tion from the firm, the nature of the software products business does not permit such a narrow vendor-buyer relationship. Other industries also have a natural continuing relationship with the buyer providing maintenance, such as the automobile industry, but this appears to be of a fundamentally different character than the continuing relationship defined by the software vendor-consumer. Both an automobile and software are presumed to be working properly when received by the purchaser. Unfortunately in both cases this may not be true. To that extent the responsibility for maintenance is similar. But whereas consumers recognize that automobiles will require repair over their lifetime, software product consumers expect that no repair should be required. Nevertheless it appears to be a fact of production that software products cannot have all of their "bugs" removed during development. Therefore the company must expect to maintain the product over time at their expense. They must be careful to distinguish between maintenance for which they are responsible versus maintenance due to the customer versus enhancements beyond the product's original purpose. Consequently the software products firm must be sure that the continued expenses are outweighed by revenues.

RESULTS OF THE INTERVIEWS

"If you were IBM we would buy it from you immediately."

This section presents the data which were collected during our interviews. It is organized into six major headings--company background, product lines, methods of advertising, customer profile, development practices, and support mechanisms. The analysis and conclusions in the fourth and fifth sections are based in part upon the data which we present here.

Company Background

If we divide the companies we interviewed on the basis of gross annual revenues, then we had seven companies which gross less than $2 million per year, seven companies which gross between $2 and $20 million per year, and seven more companies which gross over $20 million. A list of the companies is seen in Figure 1. Eleven of the companies had district offices scattered around the country while the other ten essentially had one office. In general the companies with only one office had the smaller revenues, though several of these companies grossed over two million dollars last year.

Less than $2 million	Between 2-20 million	More than $20 million
Appl. Software Inc.	Compuserve Inc.	Applied Data Res.
Caine, Farber & Gordon	Computer Associates	Auto-Trol Tech.
Dylakor	Cybertek Comp. Prod.	Boole & Baobaye
Evergreen Consulting	Int'l Management Systems	C.A.C.I.
Forth Inc.	Progeni Systems Inc.	Information Inc.
Occidental Comp Sys.	Proprietary Soft Sys.	M & S Computing
PMS Systems Inc.	Tower Systems	United Comp. Inc.

Figure 1. The Interviewed Companies

Looking at the number of employees per company, we find that the small companies had an average of thirteen employees, while the medium sized average 112, and the larger ones averaged 757. To no one's surprise, we see a strong correlation between revenues and number of employes. As computer software development is known to be a labor intensive activity, growth is tightly coupled to labor costs. In today's marketplace this fact is tempered by the shortage of qualified personnel. We also determined the split between the development/support staff and the sales/marketing staff. The percentage of people involved in development/support went from a high of 62% for small companies, to 65% for medium, to 53% for large. The sales staff was typically 10-20% of the entire staff with the remaining group comprised of marketing and business people such as accountants, personnel administration, etc.

There was a potpourri of questions we asked about the companies which produced an array of facts, many of which give some insights into the industry as a whole. For example six of the companies were publicly owned corporations, but none of the others had any immediate plans to go public or felt that there was any great reason to do so. Most of the companies we interviewed were founded between 1967 - 1970, three were founded before then, and six after 1970. Most of the companies were started by technical people and began with a staff of three or four. Typically these firms did contract programming to maintain themselves, while they attempted to build up sales of their software package.

Our sample shows this industry to be populated by many young and small companies which usually have been founded by technically oriented people. During the start-up phase they must find a means to support themselves until the product begins to generate sales on a continuing basis. Contract programming is one solution to this problem. Often the software product had already been developed in another context and then the entrepreneurs stepped in, hoping that by providing marketing and maintenance they can successfully sell the package.

The larger companies rely on a marketing force which is nationally distributed in major cities. Overseas connections are not unusual but still form a minor part of the company's revenues. Though the sales staff of larger companies formed an increasing percentage of the overall staff, it continued to constitute less than half of the entire work force.

Since at least 50% of the employees are involved with the development and support of the software and as the labor market for these people is scarce, their salaries constitute a major portion of the company's expenses. The typical marketing/sales staff is 11% for small and medium companies, but for large companies a jump to 36% occurs in our sample.

The Product Lines

Of the twenty-one companies that were interviewed, thirty-one software products were surveyed. Figure 2 categorizes the products into one of three types: systems, utilities, or applications. These three categories have become a standard taxonomy for software products. We feel that they provide too gross a splitting of products and we will address the question of a better taxonomy in the fourth section. We see that more than half are classified as application products.

Data were collected for what computer(s) the products run on. It is no surprise that a large percentage of the products run on an IBM machine or on an IBM compatible machine such as an Amdahl or Magnuson. Figure 3 shows the number of products which run on a particular machine. As a single product may run on several ma-

Type of product	Number of packages	Percentage
Systems	6	19%
Utility	12	39%
Application	16	52%

Figure 2. Distribution of Products Into Three Types

Machine type	Number of packages	Percentage
IBM	27	87%
DEC	6	26%
CDC	4	13%
Burroughs	5	16%
Honeywell	4	13%
Mini/Micro	6	19%
Others	6	19%

Figure 3. Distribution of Products by Machine Type

Target	Number of packages	Percentage
Industry only	5	16%
Mqchine only	15	48%
Industry & machine	7	23%
Neither	4	13%

Figure 4. Distribution of Products by Industry or Computer

chines, the sum of the percentages exceeds 100.

We asked if their product line was targeted at a specific type of machine or to a specific industry. Figure 4 shows the results. Almost half of the products were aimed at a specific machine. Of those aimed at a particular machine, 95% were targeted toward IBM equipment.

Another question was how the product was developed, either inhouse or by acquisition or via a programming contract. Figure 5 shows the results. It was expected that most software products would have been developed under the auspices of an outside software programming contract, so it was a surprise to find that almost as many products got started from inhouse development. Acquisition is another way for a company to obtain new products which offers many advantages. Chief among these is the fact that a product already exists and has a user base. This approach however is tempered by the acquiring company's desire to focus on a set of packages which fulfill a coherent set of user needs. Therefore these companies rarely choose to purchase and market products simply on the basis that they work well or that there is a need and a well-defined market.

A large percentage of the software products are sold at a one-time cost. In this survey, the prices range from a low of $2000 to a high of $200,000. Figure 6 gives the statistical breakdown.

Finally, information on the number of installations using a software product was surveyed. The number of installations ranged from a minimum of six to a maximum of 7000. The breakdown is given in Figure 7. In analyzing the data, we found that the number of installations is independent of the price. Moreover, the increase in the price of a software product did not affect its rate of sales. A product failed to meet sales expectations either because users were not interested in it or because a superior product existed.

To a large extent existing hardware continueds to be the determining factor in the software products market. As the installed base of IBM machines is so much greater than for any other computer, vendors commonly elect to develop applications on an IBM or a plug compatible computer.

Advertising

In this category we tried to determine what the companies were doing to get the word around about their products. Figure 8 shows the different ways that the companies advertise. Without exception our interviewees confirmed the fact that sales is still dominated by referrals and personal contacts. Advertising in national periodicals was aimed at getting the name of the product into the viewer's mind. This is based on the premise that a buyer is more likely to purchase a product whose name is recognized. Advertising the name of the company is aimed at establishing a good reputation for the firm, a critical ingredient in a software product sale. However, advertising of this type rarely led to actual sales.

We asked the companies what they regarded as their major marketing asset. Marketing knowledge, company size and resources, uniqueness of the product, or advertising were all considered incidental compared to the reputation of the company and the performance of the product.

In all cases advertising accounted for less than 4% of their operating budget. However, during the promotion of a new product advertising was notably higher than for older products. Other miscellaneous facts were that sales offices were typically small, around three to five people. The sales staff were paid on a salary plus commission basis with commissions ranging from a low of 5% to a high of 22%

How software got started	Number of packages	Percentage
Contract	13	23%
Inhouse	12	39%
Acquired	6	19%

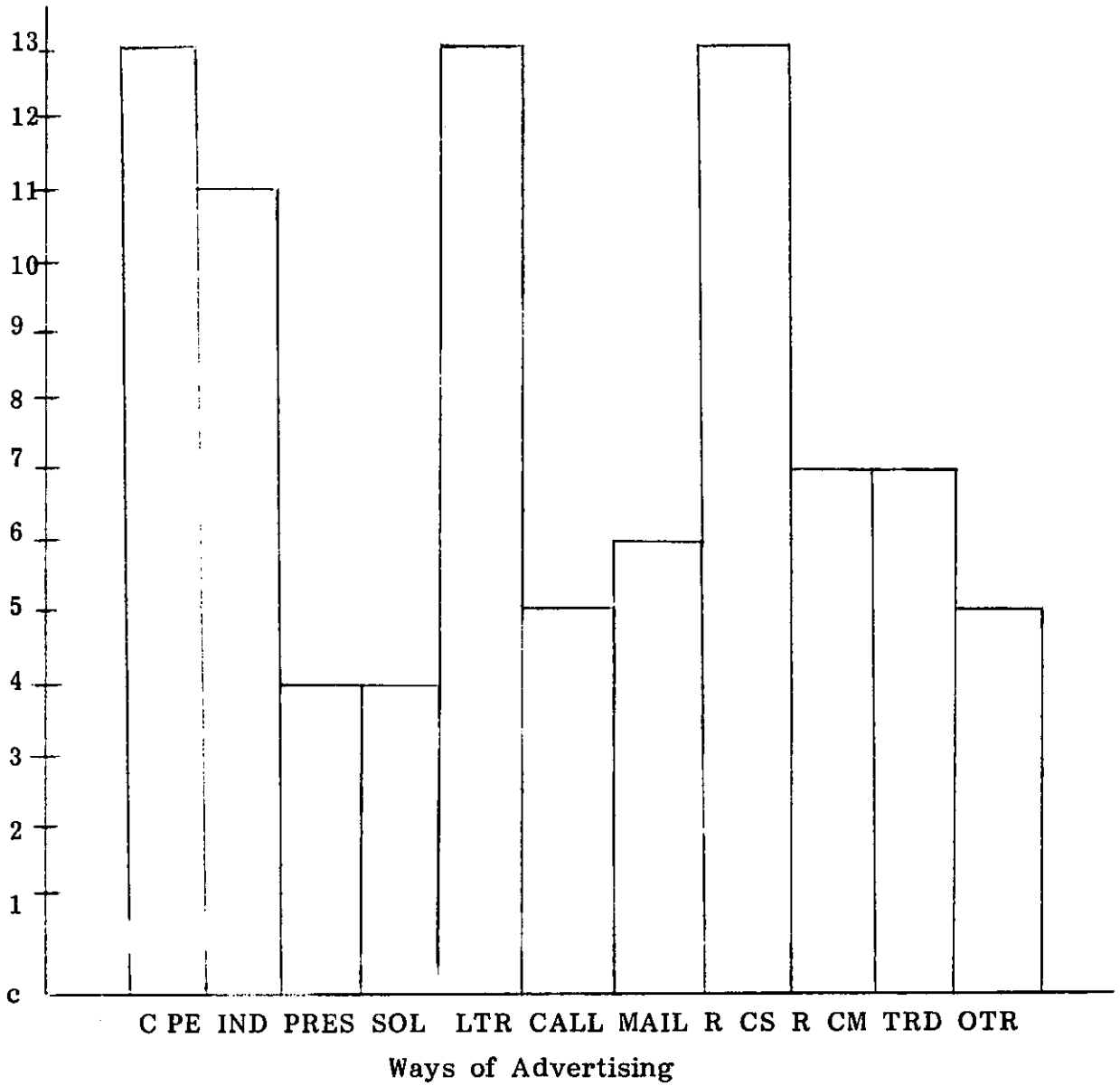
Figure 5. How Software Packages Were Initially Developed

Price Range	Number of products	Percentage
$0 - $9,999	7	23%
$10,000 - $24,999	5	16%
$25,000 - $49,999	12	39%
$50,000 - $99,999	3	10%
$100,000 - $200,000	2	

Figure 6. Distribution of Products by Price

Number of Installations	Number of packages	Percentage
0 - 49	8	28%
50 - 99	2	6%
100 - 249	6	20%
250 - 499	6	20%
500 - 999	2	6%
1,000 - 2,000	5	16%
Over 2,000	2	6%

Figure 7. Distribution of the Number of Installations Per Product.



C PE	Computer Periodicals	R CS	Referrals from Customer
IND	Industry Specific Periodicals	R CM	Referrals from Company
PRES	Presentation	TRD	Trade Show
SOL	Solicitation	OTR	Others
LTR	Letter	O IN	Outside Industry
CALL	Telephone Calls		
MAIL	Purchased Mailing LIst		

Others includes: outside industry, LA TIMES, classified ad, describe products in conferences.

Figure 8.

Customer Profile

The independent software package market as defined here does not restrict its sales to any particular group. However as the price of their products are upwards of $10,000, this fact alone tends to define in a major way the type of customer one sees. Traditionally the customer is a corporation which has a data processing department using IBM machines and a yearly budget over $1 million.

In the survey the price of $10,000 seemed to be a point of departure. Packages costing less than this amount were typically purchased by a technical person, while packages costing more, generally involved management at a higher level. Within the software product companies, people with a technical background preferred to deal with technical people while people with a management background preferred to deal with management. The "two cultures" still exist. Though the typical user is described as experienced with computers, he may not have purchased a software package before. Nor may he have a formal mechanism for determining which package to purchase. Some larger companies often have one person assigned to coordinating the task of software evaluation and purchase.

It was often reported that the most successful approach to selling followed this sequence: (1) contact the DP manager and get his approval to talk to his staff; (2) win over the staff by convincing them of the desirability and technical superiority of the product; (3) have them encourage the DP manager to make the purchase. This "algorithm" is strongly supported by statistics gathered by the Datapro Research Corp. who report in a recent survey that 80.9% of their 2,140 respondents say that their DP manager approves all software package acquisition and 90.3% say that he participated in evaluating outside software packages (Datapro 79).

Development Practices

In this category we found that the companies predominantly use IBM machines for developing their own products. Even the companies which were producing products which run on several machines used IBM machines for development. The second most common machine for development was made by Digital Equipment Corporation. Those companies which developed products for many machines would typically use a customer's machine to perform the transfer. Seven of the companies did not own their own computer, but rented computer time.

Regarding development practices, there were some surprises as well as some expected results. The question about whether quality assurance testing was done by a separate group was split about evenly (six responding yes and seven no). Smaller firms tended not to have this function as a separate activity. Similarly, on the question of whether a separate documentation staff exists, six said yes and ten said no. Although this result was in part due to the small firms, several of the larger firms surprisingly did not see the importance of a separate documentation group. The use of special development tools was specifically mentioned by only two firms. Seven firms reported using structured coding and other formalized coding methodology, while five firms were specifically not utilizing any of these methodologies. Five of the twenty-one firms reported that they had no special management hierarchy for program development such as chief programmer teams. Of the eight firms which reported the development language used, five indicated assembly language, one COBOL, FORTRAN, and RPG.

A third of the firms reported that customization was done on at least one of their products. Surprisingly, a third of the firms reported that user modifications were permitted. For those who did not permit such

modifications, the contract typically specified that the product would not be supported if the user tampered with it. While object code was the standard form of distribution, a surprising one-third of the firms reported that the source code was distributed in at least some cases. Finally, three firms reported that special hooks, or self-destruct code, was used to prevent the products from being used in an unauthorized manner.

Every company we interviewed used copyrighting as a means to protect their software. This means that both their documentation and their code contains a copyright message. In addition they make every effort to maintain secrecy, so as to qualify their product for trade secret protection. This dual approach has recently come under criticism from a group of lawyers, who suggest that the copyrighting of a work implies that trade secret protection is no longer applicable. There is an area which will likely come under greater scrutiny in the next few years.

Support Mechanisms

Most of the companies termed maintenance as keeping the program running up to specification on the most recent version of the computer it is designed to run on. Small enhancements are usually included but major enhancements are termed upgrades and incur additional cost. One-third of the companies provided some form of "hot line" to answer questions on an immediate basis.

Companies do not sell their products outright. Instead they license the use of their products through two types of leases. The first type is a permanent lease and requires a one time fee. This fee is sometimes referred to (incorrectly) as the purchase price. The second type is a rental lease which typically runs in monthly increments. We found these two agreements are used in equal proportions. The permanent lease typically includes one year of maintenance while in the rental agreement the cost of maintenance is included. Maintenance on a purchased product is usually charged as a percentage of the purchase price and varies between 10-20 percent per year. The product is generally sold for one site with discounts for multiple sites. Occasionally the monthly lease contract includes a provision for later purchase. In that case a percentage of the total amount paid on leases will be applicable toward the purchase.

Most of the companies provided for user training. The terms are set forth in the contract. The time may vary from a half day to one week. The training is usually at the customer's site, but some companies do offer courses at their own offices.

A newsletter is typically maintained by the company to alert users to problems and new enhancements. The more successful products develop user groups who may meet on a regular basis.

We received eight contracts which were examined. The following clauses were present in all of them: cost of lease/purchase, length of lease, list of materials to be delivered, machines which the product runs on, backup site allowance, who can use the product, copying of materials, patent and copyright indemnification, governing law, liability limitation, warranty, maintenance and loss specification, and finally the type of protection: license, proprietary information, trade secret, copyright, patent. Because it is in the nature of the software product that it must be fully operational before it can be judged to be effective, there is a vast potential for disputes to arise. Under such circumstances, the written contract becomes the focal point for resolving any differences.

This concludes the summary of information which resulted from our interviews. In the next two sections we try to merge all of

this information to give a coherent picture of the industry as a whole.

A TAXONOMY FOR THE INDUSTRY

One obvious conclusion about this industry is the fact that there is great product differentiation. However, even within a specific product type there may be a tremendous range of functionality and price. These differences in functionality arise in two major ways. The first way would include specific capabilities or performance characteristics which a product exhibits. The second way has to do with the manner in which the product is used by the human, the so-called man-machine interface. This interface can be a very significant factor in the overall quality of a software product. But the differences in the man-machine interface make it extremely difficult to compare products. If the concepts, terminology, and methods of use employed by one product differ substantially from another, and they often do, then easy comparisons are severely hampered. It is these two dimensions of product distinctness which makes the purchasing of software packages such a challenging and interesting area of research. We found this issue to be out of the range of our survey and hence we will say nothing further about it here. However, we found it especially useful to try to place the existing products into separate categories and this is what we discuss in this section. Such a categorization can lead to a better understanding of the dynamics of the industry as a whole, and potentially provides a manner in which one can grasp the consequences on marketing of choosing to build a product within a given category. But even the creation of a product taxonomy has its difficulties.

In the trade literature it is common to find that software products are divided into three categories called systems, utilities, various products as they complement and enhance existing IBM systems. To be more specific, consider the product called a teleprocessing monitor (or TP-monitor for short). Figure 9 shows eleven competing TP-monitors and their developers. TP-monitors are a highly competitive arena and an expensive product ($30,000 - $50,000). What does this software do? Basically it permits a host of remote terminals to be attached to a mainframe computer so that terminal users can interact with their programs in realtime. The leading TP-monitor product today is IBM's CICS, and among the competitors WESTI from Westinghouse Electric holds the major share (Shoor, 1981). Getting back to the taxonomy issue, we might ask "to what category does TP-monitor belong?" In the IBM world, a TP-monitor is not part of the operating system, but is an enhancement to it. To some, this implies it should be viewed as an application product. However, we observe that for other mainframe manufacturers (such as Digital Equipment Corp.), the capabilities of a TP-monitor are routinely included in the operating system. Examining another source, the computer science literature, we find that the capabilities of a TP-monitor are discussed in either a systems programming course or an operating systems course. Therefore we have concluded that for any taxonomy to be long lasting and applicable to the widest range of products, we should ignore the influences of any particular manufacturer. The following set of six categories tries to accomplish these goals.

After studying the broad range of products we have concluded that it is fruitless to search for a set of mutually exclusively categories. We have contented ourselves with a taxonomy which divides the space of products into almost distinct categories, however admitting that some products will continue to logically reside in more than one place. We now present a set of six categories which we found to be very useful when attempting to examine the entire

TP - Monitors	Vendors
CICS	IBM
Com-plete	Software A.C. of North America
Datacomm/DC	Applied Data Research
Environ/1	Cincom Systems Inc.
IDMS/DC	Cullinane Database Systems Inc.
Intercomm/Minicomm	SDA Products
MPG SWIFT	Mathematica, Inc.
NTCS	IBM
SHADOWII	Insac Inc.
Task/Master	TSI International
WESTI	Westinghouse Electric

Figure 9. Some Names of Teleprocessing Monitors and Their Manufacturers

and applications. There seem to be no satisfactory definitions of these terms. Though there seems to be general agreement about what fits into the systems category, the distinction between utilities and applications is vague. Moreover, the category applications is far too general. For example this category includes database management systems (DEMS), general ledger systems, inventory control systems, program development systems, and in short, everything which is not included in either the systems or utilities categories. Is a routine which transfers a file from a disk to a tape a systems program (as it is usually supplied by the manufacturer), or a utility (as it is not part of the operating system), or an application (as it is used by the end-user)? It is clear to us that this trinity of category serves little purpose and a more detailed taxonomy is desirable.

One major impediment to creating a taxonomy is the fact that the software products industry is heavily oriented toward IBM equipment. As a consequence, many people within the industry understand the marketplace of products. For each category we provide a definition and list several products which belong to the category. We also discovered that there exist several marketing characteristics which are shared by all products within a given category. These also will be presented here.

Systems Software

This category consists of the software whose purpose is to control the use and interaction between the resources of the machine. These resources include the central processing unit, the tapes, disks, remote terminals, printers, punches, etc. It includes such program products as the operating system, job scheduling, file management systems, device controllers, and teleprocessing monitors.

There are several factors which these software products have in common. One is that they are highly used pieces of software and so they must be extremely reliable and very efficient. If a software products firm is trying to compete with the mainframer in this category, then he must face several facts. First, his sales staff must have high technical knowledge of all competing products. As the people they will deal with will likely be the systems

programming staff in the Data Processing Center, this reinforces the need for in-depth technical knowledge. Although the market is potentially as large as the number of installations of mainframes, this can be misleading. In the case of operating systems, a computing center is very reluctant to give up the manufacturer supplied operating system. Only in very rare cases has a competing operating system been widely adopted. The example which comes to mind is UNIX, developed at Bell Telephone Laboratories for the Digital Equipment Corporation PDP - 11 line of computers. Therefore for some systems products the market will be very small, while for others it will remain large. With respect to training and customer support, this can vary widely for products in this category. Operating systems may require extensive training and invariably require extensive field support. On the other hand a system back-up facility may involve no formal training and no contact with the customer following the sale. Another important point regarding system software focuses on the need to perpetually modify the software whenever there is a change in the machine interface software. The most common example of this phenomenon is when changes to the operating system require modification to all other systems software. However, to the extent that a particular piece of system software does not interact intimately with the operating system, one may be at least partially insulated from the need to make this type of frequent change.

Computer Operations Software

This category consists of the software which is employed by the computing center support staff to aid in their task of running the computing center as efficiently as possible. Products which fit into this category include systems for performance measurement, job accounting, installation verification systems, and various software monitors. There are many attractive aspects to marketing in this product line. The potential market is large and the mainframers do not necessarily provide competing products. To sell the product one would typically deal with the DP manager and his staff. The software products themselves are relatively small and require only a low level of field support. However they are also tightly coupled to the operating system and hence may be subject to some revision with each operating system update.

Data Management Systems

This category of software includes those systems which deal with the formatting, storage, and manipulation of logical data. Products in this class include DBM's, data capture systems, data presentation systems, data retrieval systems, data security systems, and database communication systems. For products in this category, the software firm's sales staff must now deal both with DP managers and functional managers within the organization. These people often have different criteria for package selection, thus complicating the selling process. The relative importance of an easy-to-use query language being a case in point. Also, the sales staff must bring to their task a combination of technical knowledge and knowledge of the way the product will be used by the company. Once sold the training component is often high. The attractiveness of marketing products in this category is that the market is large and the price tags are high. However, these circumstances have made this a highly competitive arena for software products.

Software Development Tools and Systems

The category consists of software systems which exist for aiding the development of software systems. It includes such pro-

ducts as language compilers, program design languages, documentation aids, software management and control systems, application generators, test data generators, and software cost estimators. The major factor which influences the sale of products in this category is the high learning curve which exists. Many of these products embody a methodology for constructing software. The methodology then must be sold to the organization before it will purchase the tools. The products are typically sold to project leaders in the environment of large software development. Despite the fact that there are many places which engage in this activity, the high learning curve causes the market for these products to be modest.

Functional Applications

This category consists of software systems which perform generic tasks found in many organizations. Chief among these are the inventory control systems, payroll programs, general ledger systems, and business forecasting systems. Also included are the packages for engineering functions such as numerical analysis packages, statistical packages, and linear programming systems. Here we have a less computer sophisticated buyer, who is normally not associated with the DP center. For business applications software there is pressure to customize each package as firms are reluctant to alter their mode of operations to fit the software package.

Industry Specific Systems

This final category is self explanatory. We found that the industries for which the most software products are currently available are: the banking industry, the life insurance industry, the mining industry, manfacturing, and the medical industry. The advantage of this type of product line is the ability to readily identify the potential customer base. Second is the possibility of devising a set of products which ideally fits the industry segment. Typically products in this category consist of a base product with several options which is then customized for each buyer. The customization is done by the package vendor and the costs are usually recovered from the customer. A factor which can inhibit the marketplace is if the industry is not yet automated to a significant degree. In this case a turnkey system may to be more appropriate.

CHARACTERISTICS OF THE INDUSTRY

"With a growing list of products, 10 years of experience, and a growth rate of 50% or better for each of the last four years, XYZ corporation is typical of the well established software vendors."

In this section we try to describe the industry as a whole. We first present the broad picture of the size and scope of the industry. Then we offer a categorization of the participating firms into three distinct groups and show how the groups interact with one another. Then we document those facets of the industry which place it on an early stage of the industry life cycle. We conclude with some prognostications about directions the industry may take in the future.

SIZE AND SCOPE

It would be useful if we knew precisely the number of firms which comprise the computer software products industry. However, there appears to be no easy way of determining this number exactly. One mechanism we used was to examine the ICP Directory which lists software products, their function, price range, and operating environment (ICP, 1980). Their 1980 directory contains the names of 617

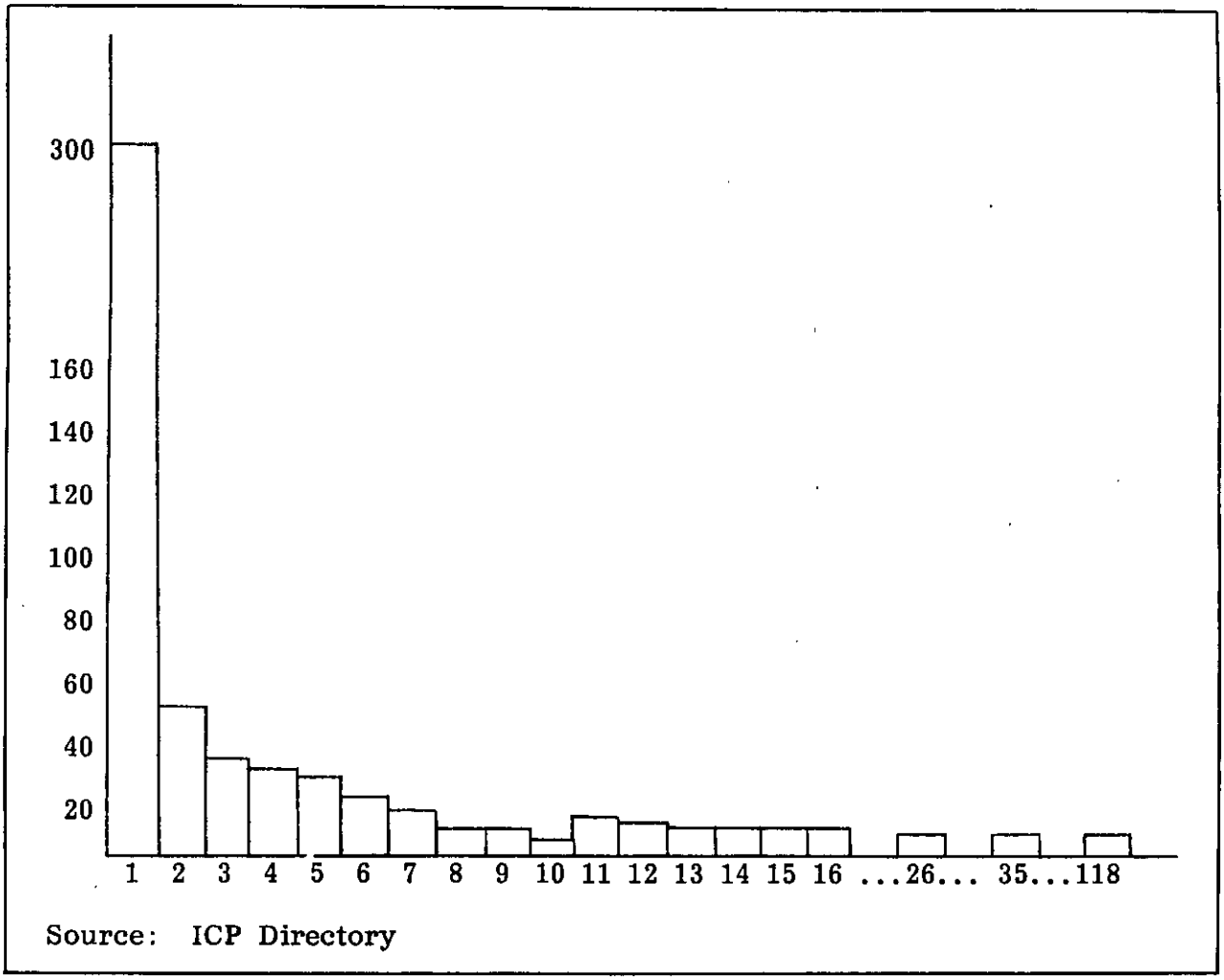


Figure 10. The Number of Companies Having 1, 2, 3, ... Products

firms selling 1676 products. Figure 10 shows the distribution of the number of products per firm.

Pursuing another source of evidence we examined the Datapro research reports, (Datapro 1979). Figure 11 shows the number of packages which were rated for the years 1976-1979, the number of vendors who supplied these packages, and the number of firms whose packages were rated for the first time in a given year.

As we can see from Figure 11, the number of packages has grown to approximately 2000. We also observe from this figure that each year approximately 25% of the firms are new to the rating scheme. Some more arithmetic shows that a similar percentage of firms are no longer on the list which had been there in the previous year. Furthermore, we examined the number of products offered by each vendor as rated in the Datapro reports. Of the 140 firms, eighty-three had only one rated product, twenty-four had two products, six had three and five products, eleven had six products, and one had seven, eight, nine, ten, seventeen and sixty-four products, the latter being IBM. These statistics should

Year	Number of packages	Number of Vendors	Number of New Firms Rated
1979	2142	120	30
1976	1961	100	32
1977	1223	79	20
1976	1446	104	35

Figure 11. Summary of Datapro Statistics on Vendors and Packages

be compared with Figure 10 as they lead to similar conclusions.

Using the results described in the previous paragraphs, we conclude that the industry consists of at least 1000 firms selling more than 2000 software products. We emphasize that these numbers are best interpreted as lower bounds to the true figures. Oakey Mertz, from the Cartner Group (a major IBM watcher) estimates that IBM alone has 2700 program products (Mertz, 1981). Therefore we believe that the true number of software products is substantially larger. Given that directories such as ICP 1980 do not include all new firms, the true number of firms may also be substantially larger.

Grouping the Firms

It is possible to organize these firms into three distinct groups, where the participants in a group share similar characteristics. The first group consists of the computer mainframe manufacturers. These companies have always provided approximately 25% of the firms are new to the rating scheme. Some more arithmetic shows that a similar percentage of firms are no longer on the list which had been there in the previous year. Furthermore, we examined the number of products offered by each vendor as rated in the Datapro reports. Of the 140 firms, eighty-three had only one rated product, twenty-four had two products, six had three and five products, eleven had six products, and one had seven, eight, nine, ten, seventeen, and sixty-four products, the latter being IBM. These statistics should be compared with Figure 10 as they lead to similar conclusions.

Using the results described in the previous paragraphs, we conclude that the industry consists of at least 1000 firms selling more than 2000 software products. We emphasize that these numbers are best interpreted as lower bounds to the true figures. Oakey Mertz, from the Cartner Group (a major IBM watcher) estimates that IBM alone has 2700 program products (Mertz, 1981). Therefore we believe that the true number of software products is substantially larger. Given that directories such as ICP 1980 do not include all new firms, the true number of firms may also be substantially larger.

Grouping the Firms

It is possible to organize these firms into three distinct groups, where the participants in a group share similar characteristics. The first group consists of the computer mainframe manufacturers. These companies have always provided at

least a minimal set of software with their machines. However, the extent to which they provide software beyond the systems category varies widely. In the past most of the manufacturer provided software was included with the purchase price of the hardware. In 1969 IBM made the decision to separately price its software from its hardware (called unbundling). Other manufacturers have followed suit. This step gave a big boost to the growth of this industry. The mainframe manufacturers represent a significant force in the industry. Figure 12 shows the top ten mainframe manufacturers, their revenues from software products and services (SPAS), and their total revenues. This information was obtained from ICP 100.

The next group consists of those companies which have several products, annual revenues in the tens of millions, a sales force which is geographically distributed, and low profitability and high growth. In addition, these firms typically have their own computer system (an IBM machine) and at least one highly successful product for which the company is well known. Many of these firms also engage in other data processing activities including contract programming, consulting, facilities management, and network services. Figure 13 contains a list of the top firms in this category.

The final group of firms is characterized by the fact that they have essentially one product, less than ten people, most of whom are engaged in technical activities, none or a small sales force, and annual revenues less than two million. This group includes the vast majority of firms in the industry. From Figure 10 we see that one product firms constitute one-half of the 617 firms, while firms with less than four products make up two-thirds of the sample. It is these firms which are engaged in the high risk/high return aspect of the soft-

Company	SP&S Revenue (millions)	Total Revenue (millions)
IBM Corp.	$1,835	$26,213
Honeywell Inc.	205	4,925
Sperry Corp.	383	4,785
Control Data Corp.	1,036	3,800
NCR Corp.	598	3,322
Hewlett-Packard Co.	305	3,099
Burroughs Corp.	580	2,902
Digit. Equip. Corp.	589	2,368
Perkin-Elmer Corp.	141	996
Data General Corp.	105	654

Figure 12. Largest Computer Manufacturers in the SP&S Business

Source: ICP 100, 3rd Annual Survey, International Computer Programs Inc.

Company	1980 Soft Revenue (millions)	Total Revenue (millions)
Management Science America	$48	$ 52
Informatics	40	100
Cincom Systems Inc.	34	37
Policy Management Systems	32	46
Pansophic Systems Inc.	31	31
Applied Data Research	29	37
American Management Systems	29	59
Software AG of North America	26	28
Computer Associates	25	25
University Computing	20	82
Kirchman Corp.	19	32
Cullinane Corp.	18	18
Software International Corp.	17	17
McCormack & Dodge Corp.	16	16
Information Science Inc.	16	16
Insurance Systems of America	14	22
Rand Information Systems	14	20

Source: ICP 100 3rd Annual Survey, International Computer Programs Inc.

Figure 13. Largest Software Suppliers

ware products business. The few firms who develop a successful product will realize tremendous profit, but as in the record and movie business, only a small number of these products will become "hits."

Figure 14 shows in pictorial form the three segments of the software products industry. As far as we can determine, the industry has a significant (approximately 25%) number of firms which enter and leave each year. Some of the one product firms are acquired, while others create various kinds of liaisons with the larger firms. In the past the mainframers relied almost exclusively on their inhouse staff to produce program products. This appears to be changing as we discuss in a later paragraph of this section.

Regarding the acquisition of smaller firms by the larger ones, the motivation for this behavior works in both directions. Typically the smaller firms are strongly oriented toward their product's technical aspects. As a result they frequently are lacking in the areas of marketing and distribution. Also they lack an established reputation which is so important for selling products which are difficult to evaluate. Recognizing this, the smaller firms may seek some type of relationship with the larger entities. The larger firms are motivated to acquire a product rather than do development inhouse because they can thereby avoid the large risks and difficulties associated with new product development. Extrapolation of this line of reasoning would argue that the larger firms may soon

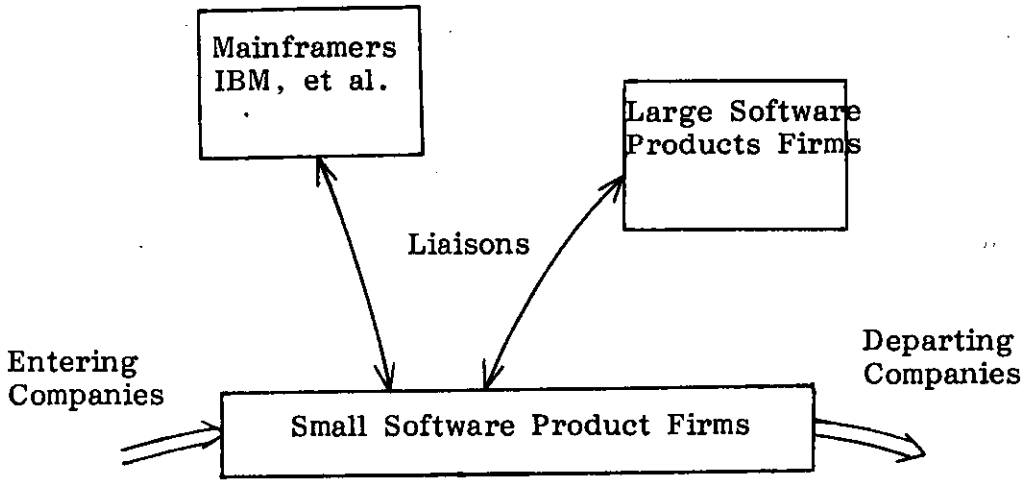


Figure 14. The Three Groups Forming the Software Products Industry

become primarily conduits for products developed by other people. The analogy with the publishing industry should not be overlooked.

Industry Life Cycle

A new industry is born when a person or group of people perceive a need and establish a "going concern" to provide goods or services to fulfill the newly identified need. The birth of the software products industry could be placed somewhere near the early or mid 1960's with one of the first big successful products being Informatic's MARK IV. In the late 1960's investment captial helped establish several new firms. In life cycle terminology this was the embrionic period. The next phase, called the growth period, is marked by impressive growth in which everyone benefits from the expanding demand. This period is followed by the shake-out period where the number of competing firms is greatly reduced during the process of establishing the rules of competition. The final phase of the life cycle is the mature period where the demand is relatively fixed and the industry becomes vulnerable to substitute industries based on other technological forms.

Within the life cycle structure the software products industry must be placed in the growth period. The earlier figures emphasize the rapid pace at which new firms are entering the marketplace. This is supported by a recent shift in the attitude of the investment community toward firms in the software products industry. Until recently the industry was viewed as an exotic, high risk area, and hence it did not receive serious attention. However, with the realization that the lack of software is a major inhibiting force in the adoption of computers, the investment potential of the industry has become apparent. Indeed Alex Brown & Sons, an investment banking firm, has claimed that the software products industry is third only to the oil and mining industries in terms of financial leverage (Brown, 1981). Thus, we expect to see the financial community funding these companies to a far greater extent than ever before.

Using the framework developed by Michael Porter in his book entitled Competitive Strategy: Techniques for Analyzing Industries and Competitors, (Porter, 1981), the software products industry would be described as an emerging and fragmented industry. At present there are still no "rules of the game." Successful strategies involve a multiplicity of product and marketing practices. Product strategies are as varied as the taxonomy of the fourth section. Marketing practices vary in the composition of staff, amount of advertising, channels of distribution, and pricing policies.

A relatively new development is the changing attitude of the mainframe manufacturers toward the other firms. In the past, the mainframers mostly shunned software developed out-of-house (the Not Invented Here syndrome). However, as a result of market forces, the mainframers have recently changed their position. These market forces are the labor shortage and the need for applications software to sell computer systems. Another factor is that some new firms are capable of creating highly innovative products while the larger companies are forced to adopt a more conservative policy toward research and development. Thus, they now recognize that they cannot afford to ignore this potential avenue of new successful products. Many mainframers have even established formal programs by which software product entrepreneurs can offer their work to them via a royalty or other type of agreement. Both the mainframers and the large software product firms are now viewing the large number of small firms as a fertile arena in which new ideas can be developed and tested.

One ever present factor is the role of IBM. First of all they are an important participant by virtue of the number of products they offer. IBM software products continue to achieve the largest number of users in a broad range of fields. Though IBM does not publish revenue figures from software distinct from hardware it is believed by industry observers that for 1980 software revenues were approximately $1.3 billion, or about 5% of its total computer revenues (Mertz, 1981). This makes IBM larger by a factor of twenty than any of the firms in the other groups. There is no doubt that the relative cost of hardware, i.e., cost per function, has been and is continuing to go down. We believe that the percentage of revenues coming from software versus hardware will continue to increase for IBM. Thus, we must assume that IBM will pay increasing attention to its role as a software products firm. IBM's greatest advantage in this arena comes from their reputation in the industry. But this is a factor which the other firms must deal with even now.

A second role for IBM results from the fact that their computers are the predominant machines on which software packages are designed to run. IBM also gains from the fact that its internal software development groups have advance knowledge of the architecture of future machines or changes to existing ones. Thus, they have a head start in producing products for these new environments. However, it is not clear to what extent this is an advantage. Even Goetz concedes that the percentage of software which directly interacts with the computer hardware is very small (Goetz, 1981). And even IBM cannot afford to make radical changes in system architecture, as their own customer and software base could become eroded as a result.

Though IBM is the leader in terms of annual revenues, no single firm dominates in all product areas. Besides the mainframers, for which we cannot determine the percent of their revenues due to software sales, all of the big companies are earning less than $50 million per year. There is a lot of competition from smaller and newer companies. Figure 10 implies that 25% of the companies are leaving the

field each year, while a new 25% are replacing them. This in part is due to the relatively low investment required to enter the business. Because the new company has typically built the software product prior to entering the field, the front-end costs have already been spent out of a different budget. The subsequent marketing and support costs can be both modest and capable of being spread out over time. This minimizes the amount of capital a new firm must have in the first few years. Another factor which aids the small firm and shapes the entrepreneurial character of this industry is the fact that being a large firm does not significantly improve one's ability to build a successful software product. In fact, it may be an advantage to have a small firm consisting of a few highly bright and motivated people. Therefore we conclude that since the industry is still dynamic in terms of the participats entering and leaving the field, it is a fragmented industry.

Other signs that this is an emerging industry is the fact that there is great uncertainty about which products will ultimately be the best, and that in many cases buyers are purchasing software packages for the first time. For example if one looks carefully at the taxonomy presented in the fourth section, one sees that it is virtually impossible to make consistent marketing strategy based upon producing products in a single category. This causes great difficulty in doing long range product and marketing planning.

One pattern which is clearly forming for the larger software product companies is the integration of their product line. In the past many of these firms have either stuck to one product, enhanced in a variety of ways, or they have obtained additional products in a somewhat random manner. Once a product is built and becomes successful in the marketplace, there is a tendency to maintain and enhance it in the hope of generating further sales. This step will often be undertaken instead of pursuing new products. The tendency now is to examine their customer needs, their marketing potential, and their existing products with the result being a strategy for new product development. These products will usually offer a "complete" solution to a major data processing problem. Another form of integration is the creation of a uniform mechanism for using a set of products. Both of these steps are being taken by several of the larger firms. The advantage of product integration is that it reduces sales costs and creates additional demand. If the customer is a repeat buyer, less convincing on an additional product about the critical issue of support will be required. Second, the customer will recognize the savings in terms of costs if the package is easier to train people on, as it uses similar modes of access as other packages from the same vendor. These facts will bias the customer toward the same vendor even if the product itself is not the best in terms of efficiency or lacks a minor functional capability. Thus, through product integration the vendor experiences simultaneously a lower sales cost per unit and a larger demand. IBM has certainly benefited from this phenomenon.

Another pattern on the horizon is the growing market for software which runs on microprocessors. This market is very different from the one we have been discussing so far. In contrast, it is characterized by low prices and high volume. Also the customers are no longer large corporations, but are small businesses or hobbyists. The marketing is done largely through retail outlets. Nevertheless, these companies see the vast potential as some software products have already sold over 80,000 units, e.g., VISICALC. Thus, we anticipate that some firms will enter this new arena soon.

Competition from substitutes, a characteristic of more mature industries, does not

appear to be an immediate threat here. However, the software consumer has essentially two options besides choosing a software package, computer network services, and turnkey systems. Turnkey systems are good for specialized tasks where the application would not support by itself the cost of a technical staff, and where the task is performed sufficiently independent of other data processing tasks. Thus, there is no need to have the application running on the same machine with other tasks, and hence a turnkey system is a viable substitute. With the growth of local computer networks, the future holds even greater potential for turnkey systems, as a remote computer running one application may be connected to others, thereby allowing input and output data to be exchanged between machines. We have not tried to study the cost tradeoffs between a turnkey system and the purchase of an application software package for an existing computer. But we see a trend away from the one-large-machine computing center to a computing environment which supports a variety of machines. Thus, the turnkey system may soon offer strong competition to the software package industry. With respect to the computer network services industry we see their greatest success coming in the market of people who do not already own a computer. Thus, they are not directly competitive with the software products industry. There are, however, certain network services companies who specialize in industry specific applications and these companies will erode the market for packages. A study of both of these industries is warranted before we can accurately make any conclusions about their future effect on the software products industry.

SUMMARY

The computer industry is a vital segment of our economy. To understand its changing patterns of growth requires the ability to apply business, economic, and technological methodology in a complex manner. We have chosen to analyze the software products industry as a first step, because software represents the major bottleneck in the further rapid spread of computers and we believe that a study such as this one will lead to greater understanding of the role this industry can play in alleviating this bottleneck.

This paper has attempted to describe the computer software products industry at a level which is appropriate for the person wishing to understand the major determinants of inter-firm activity without having a detailed knowledge of specific firms and products. Descriptions of specific firms and products are best provided by the market research firms. Instead we have tried to characterize the major trends in the industry, by showing how the firms can be grouped and by providing a taxonomy for their product offerings. Our conclusions are based upon indepth studies of twenty-one participating firms, plus interviews with numerous people.

We began by emphasizing some important facets of software which are not widely appreciated, but which play key roles in the development of successful software product firms. Summarized here, some of these factors are:

- the high initial costs of software development,
- the difficulties of accurately estimating the time and cost of this development,
- the fact that maintenance costs can be twice the cost of development,
- the fact that the quality of the product cannot be determined before it is built,

- the fact that products embodying new concepts have a high learning curve, and thus will experience additional difficulty in their marketing.

In the third section we detail the results of the interviews. The questionnaire had been developed after several test interviews of participating firms were made. Though the sample size is not large, we believe that the results obtained combined with the interviews of industry leaders gives our conclusions validity.

Though the participants in the software products industry are easy to identify, their product offerings are very diverse and hence difficult to categorize. The trade literature continues to refer to systems, utilities, and applications software as the three major divisions. To us, this seems totally inadequate. In its place we offer six categories, their definition and some sample products. The categories are system software, computer operations software, data management software, software development systems, functional applications software, and industry specific systems.

In the final section we address several important industry factors. Some of the relevant points are:

- The industry can be divided into three groups: the mainframe, the larger software product firms, and the smaller software product firms. A precise set of criteria defining these categories is given.

- The latter group contains more than two-thirds of the number of firms in the industry.

- Due to various market forces, the mainframers and the larger software product firms now openly encourage software developers to bring them their products for possible joint ventures.

- The small software products firm is essentially a high risk/high return business with little chance for major growth, but a good chance for high profitability. The latter condition is not shared by the larger software companies which typically show profitability at less than 10%.

- Though IBM is the dominant supplier in terms of the number of products and dollar volume it does not appear to be taking a predatory stand against other firms.

- Successful product planning is difficult at best. Various essential factors are covered in the fourth section with each subarea of the taxonomy.

- Participating firms may soon move into the marketplace of software for personal computers.

- Competition from substitutes exists, but with demand increasing in all sectors it plays no major role.

The software products industry is a highly dynamic one. We believe we have presented an accurate and detailed picture so that the investment manager, new entrepreneur, or current participant will understand the environment which exists today.

REFERENCES

Bailey, J.W. and Basili, V.R. "A Meta-Model for Software Development Resource Expenditures," Proceedings of the 5th International Conference on Software Engineering, IEEE, March 1981, pp. 107-116.

Brooks, F.P., Jr. The Mythical Man-Month,

Addison-Wesley, Reading, Massachusetts, 1975.

Representative of Alex Brown & Sons, New York, to the ICP Million Dollar Awards Conference, San Antonio, Texas, April 1981.

Datapro Reports: User Ratings of Proprietary Software, 1974-1979, Datapro Research Corp., 1805 Underwood Boulevard, Delran, New Jersey, 08075.

Goetz, M.A. "A Separate IBM Software Company," Datamation, Volume 27, Number 3, March 1981, pp. 275-280.

"ICP Directory '80" International Computer Programs Inc., 9000 Keystone Crossing, Indianapolis, Indiana, 46240, 1980.

"ICP100 Third Annual Survey of Computer Companies." International Computer Programs, Inc., 9000 Keystone Crossing, Indianapolis, Indiana, 46240, 1981.

Mertz, O. Talk Given at the ICP Million Dollar Awards Conference, San Antonio, Texas, April 1981.

Porter, M.E. Competitive Strategy: Techniques for Analyzing Industries and Competitors, Free Press, New York, New York, 1980.

Shoor, R. "Large IBM Users Seen Migrating to MVS," Computerworld, March 30, 1981, p. 5.